A Labeling Scheme for Young Tableaux Spanning Representations of Permutation Group $S(N)$

C. R. SARMA, A. V. V. NAMPOOTHIRI

Department of Physics, Indian Institute of Technology, Powai, Bombay 400 076, India

Received 26 March 1999; accepted 16 September 1999

ABSTRACT: A structure-dependent labeling scheme for the Standard Young Tableaux spanning the representations of the permutation group is outlined in the present work. This scheme is used to generate the representations of a select class of permutations such as dense cycles and general transpositions of the group using minimal storage requirements. Two distinct approaches are outlined for generating the tableaux in the present labeling scheme. Detailed application has been made to two-column Young diagram representations that are extremely useful in electron correlation studies in molecules. J Comput Chem 21: 185–190, 2000

Keywords: structure-dependent labeling; Standard Young Tableaux; symmetric group $S(N)$; electron correlation studies

Introduction

Symmetric group $S(N)$ plays a vital role in many-body correlation problems because it is the maximal symmetry group for identical particle systems. As is well known,[1–3] the standard labeling of the basis spanning a representation $[\lambda] = [\lambda_1, \lambda_2, \ldots, \lambda_k]$ of $S(N)$ is in terms of the $f_{N,\lambda}$ Standard Young Tableaux (SYT) spanning it. The SYT consist of entries, $i = 1, \ldots, N$, filling the Young Diagram (YD) corresponding to the representation in all possible ways satisfying the Young rules.[1,2] Once the SYT are generated, the matrix representation of the elementary transpositions (ET), $\{(i, i+1) \mid i = 1, \ldots, N-1\}$, of the group follow using simple rules[1,2] in terms of the axial distances between the corresponding i and $i+1$. Although the rules themselves are extremely simple, implementing them leads to two major problems. First, the lack of a suitable indexing scheme for the SYT requires a search to locate the off-diagonal SYT, which may result from the ET. Second, the location of the given pair of entries defining the ET in the SYT and determination of the axial distance are also tedious problems. Any element $P \in S(N)$ is, in general, the product of a large number of ETs. Thus, run-time determination of these quantities is not feasible for generating representation matrices of general permutations.

Correspondence to: C. R. Sarma; e-mail: phcrsia@niharika.phy.iitb.ernet.in

Contract/grant sponsor: Council of Scientific and Industrial Research (India)

CCC 0192-8651 / 00 / 030185-06

A way out of these problems was suggested in a recent note by Rettrup.[4] In this note, three look-up tables were constructed to be repeatedly read during realization of matrix representation of ETs. One of these is a table to provide the link between a pair of SYTs connected by an ET. The remaining two tables are for numerical values for diagonal and off-diagonal matrix elements. Although this technique is reasonably efficient, there are some drawbacks in it. First, we need storage of these tables, each of which is of size $f_{N,\lambda} \times (N-1)$. Further, the representation of any arbitrary permutation expressed as as a product of ETs is realized using sparse matrix multiplication, techniques. Matrix multiplication is generally not the most efficient technique even if we are dealing with sparse matrices. The alternative to this is to use binary tree search method because each ET yields at most two branches. This technique was used by Duch[5] for cyclic permutations, and by Wormer and Paldus[6] for general transpositions. We, however, found this technique to be even slower than sparse matrix multiplication.

In the present note we have tried to address both the problems of indexing the SYT as well as that of efficient determination of axial distance between pairs of entries $(i, i+1)$ in a given tableau. As a first step, in this context, we note that the Yamanouchi realization[1-3] of the SYT yields a linear array specifying the row (or column) location of the entries in the tableaux. This reduces the effort in locating a pair of neighbor entries and determination of axial distances to some extent. Noting that the YD corresponding to a representation is two dimensional, the Yamanouchi array gives only a partial specification in terms of row or column indices. It is worth examining whether a more complete row plus column specification treating the YD as a two-dimensional array will be more advantageous in determining axial distances. The main effort in the present note has been to generate a linear array yielding complete information on entry location in a YD treated as a two-dimensional structure. We have done this using what we term as the Modified Standard Young Tableaux (MSYT) array. This is a linear array with N entries yielding the location of entries $1, \ldots, N$ in the SYT grid.

Use of MSYT results in two main advantages. First, the axial distance for a pair $(i, i+1)$ follows readily from the corresponding adjacent entry values. Further, the axial distance is the same in all MSYT having the same entry values. This, in turn, implies that any look-up tables we construct to read values of diagonal and off-diagonal matrix elements of ETs, need to be at most of size $N \times N$. If, in addition, we are able to generate the MSYT in an indexed form, we do not also need any link table.

The above scheme has been developed in the present note. The procedure is outlined in the next section, and adequately illustrated using examples. Two distinct methods of generating the MSYT and indexing them have been considered. One of these is a modification of a graphical technique developed recently[7] for generating the first columns of two-column SYTs. This technique has been used to generate the representations of dense cycles and general transpositions that are extremely useful in electron correlation studies.[8-11] Representations with YD of more than two columns are also important in other areas of physics, and need to be considered. We found it very difficult to generalize the technique presented in ref. 7 to these representations. It is possible, however, to adapt the Distinct Row Table (DRT) approach used successfully by Shavitt[11] in his Graphical Unitary Group Approach (GUGA) to electron correlation studies to generating the MSYT for arbitrary representations of $S(N)$. This adaptation has also been presented in the next section. A brief discussion of the present scheme is given in the Discussion Section.

Permutation Representations and MSYT

Consider a representation $[\lambda] = [\lambda_1, \lambda_2, \ldots, \lambda_k]$ of $S(N)$ having column lengths $\{c_i \mid i = 1, \ldots, N\}$ from left to right with c_i satisfying $c_1 \geq c_2 \geq \cdots \geq c_{\lambda_1}$ and $\sum_{i=1}^{N} c_i = N$. The SYT spanning the representation can then be realized as

$$[i] = [i_1, i_2, \ldots, i_{c_{(1)}}; i_{c_{(1)}+1}, \ldots, i_{c_{(2)}}; \ldots; i_{c_{(\lambda_2)}+1}, \ldots, i_{c_{(\lambda_1)}}], \quad (1)$$

where all entries belong to the set $\{s_i \mid i = 1, N\}$. The entries preceding each semicolon (;) belong to a column. The entries satisfy the standard Young rules of defining increasing sequences from top to bottom along a column and from left to right along a row of the YD. As an illustration of this notation consider the SYT spanning the five-dimensional representation $[2^3]$ of $S(6)$. We have,

$$[1,3,5;2,4,6], [1,3,4;2,5,6], [1,2,5;3,4,6], [1,2,4;3,5,6], [1,2,3;4,5,6].$$

At this stage it is convenient to define cumulative column lengths of the YD as,

$$c_{(k)} = c_{(k-1)} + c_k, \quad (2)$$

subject to $c_{(0)} = 0$. The MSYT can now be readily generated from the tableaux of eq. (1) as follows: we first note that the location (i, j) of a box in a YD can be assigned a single running index using the cumulative column lengths and,

$$(i, j) \rightarrow c_{(j-1)} + i. \tag{3}$$

Such an index, in turn, can be associated with SYT entries of eq. (1) by assigning integer entry values $(l(i_k) \mid k = 1, \ldots, N)$ successively as

$$l(i_k) = k. \tag{4}$$

As an illustration, consider the representation $[4, 3, 2]$ of $S(9)$, which has column lengths $[3, 3, 2, 1]$. Let $[1, 2, 5; 3, 4, 8; 6, 9; 7]$ be an SYT spanning the representation. Using eq. (4) we obtain $l(i_1) = l(1) = 1$, $l(i_2) = l(2) = 2, \ldots, l(i_9) = l(7) = 9$. This leads to the MSYT array, l: $[1, 2, 4, 5, 3, 7, 9, 6, 8]$. Thus, a loop of length N based on Eq. (4) converts the SYT to MSYT. The main advantage of the modified form of the tableau is that the entry values in it are purely dependent on the structure of the YD, and the location of an entry follows simply from its location in the array. No search is needed when determining the axial distance for an ET pair $(i, i+1)$. Further, the structure dependence implies that any two arrays having identical entry values for a pair $(i, i+1)$ have the same axial distance independent of the tableaux index. This permits us to tabulate the axial distances and, hence, the matrix elements of ETs in a $N \times N$ array rather than in a $N \times f_{N,\lambda}$ as in ref. 4. As an illustration of the MSYT arrays, we consider the representation $[2^3]$ of $S(6)$. Using eq. (1) for the SYT listed earlier, we obtain the MSYT arrays,

$$[1, 4, 2, 5, 3, 6], [1, 4, 2, 3, 5, 6), [1, 2, 4, 5, 3, 6],$$
$$[1, 2, 4, 3, 5, 6], [1, 2, 3, 4, 5, 6].$$

The arrays defined using eq. (4), however, are still not quite convenient for determining the axial distances between adjacent pairs $(i, i + 1)$. A slight modification to define a temporary array, $\{l1(i)\}$ as follows overcomes this problem. Each of the entries $\{l(i) \mid i = 1, \ldots, N\}$ are compared with the cumulative column lengths $\{c_{(j)} \mid j = 1, \ldots, \lambda_1\}$ until the inequality,

$$c_{(j-1)} \leq l(i) \leq c_{(j)} \tag{5}$$

is satisfied. The modified entry is then given by,

$$l1(i) = l(i) - c_{(j-1)} - (j - 1) \tag{6}$$

for $i = 1, \ldots, N$. The axial distance for $i, i + 1$ now follows as,

$$d(i, i + 1) = l1(i) - l1(i + 1). \tag{7}$$

This equation yields not only the magnitude of the axial distance but also its sign as per the Young rules.[1, 2] The justification for eq. (7) follows from the fact that this distance can be computed from the SYT by pushing the entries concerned to the extreme left column of the YD adding unity for every step taken and finally adding the difference between their row indices. The d-matrix so generated has size $N \times N$, and can be readily stored. Instead of this it is more convenient to store the diagonal,

$$\mathrm{diag}(i, i + 1) = 1/d(i, i + 1) \tag{8}$$

and the off-diagonal,

$$od(i, i + 1) = \sqrt{1 - \left(\mathrm{diag}(i, i + 1)\right)^2}, \tag{9}$$

elements of the ETs. These are also $N \times N$ matrices and can be easily stored. As an illustration of the $l1$-array resulting from eq. (6), we give below the array corresponding to the MSYT of $S(9)$ considered earlier:

$$[1, 2, 0, 1, 3, -1, -2, 2, 0].$$

From this the axial distance array follows as,

$$d\text{: } [-1, 2, -1, -2, 4, 1, -4, 2].$$

As a further illustration we present below, the $l1$-arrays for the $[2^3]$ representation of $S(6)$:

$$[1, 0, 2, 1, 3, 2], [1, 0, 2, 3, 1, 2], [1, 2, 0, 1, 3, 2],$$
$$[1, 2, 0, 3, 1, 2], [1, 2, 3, 0, 1, 2].$$

The axial distances can be readily determined from these arrays using eq. (7), and we can generate the axial distance matrix as,

$$\begin{bmatrix} 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 3 & 2 & 1 \\ 0 & -2 & -3 & 0 & -1 & 0 \\ 0 & 0 & -2 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Once this matrix has been generated, the diagonal and off-diagonal elements of Young orthogonal representation for the ET's follow from eqs. (8) and (9). Because these are now obtained in a tableaux-independent form, they can be stored as $N \times N$ matrix arrays. For a given representation, these matrices are generated once only and stored so that they do not lead to any run time losses.

As outlined in the Introduction, there are two approaches possible for generating the MSYT in an indexed form. One of these[7] is suited for two-column or two-row YD, and cannot be generalized easily to other representations. It is a graphical

approach for generating the first columns of SYT spanning a two-column, $[c_1, c_2]$, representation. This approach[7] consisted of first generating a weight diagram $\{w(i,j) \mid i = 1,\ldots,c_1;\ j = 1, \max(i)\}$ where c_1 is the length of the first column and $\max(i)$ is the maximum length of the ith row and has the value $2\times i-1$ for $i = 1,\ldots,c_2$, and continue with the same length for all the rest of the rows. Scanning this diagram sequentially, it is possible to generate the first columns of SYTs spanning the YD efficiently. Let $(i_1, i_2, \ldots, i_{c_1})$ define the first column of an SYT having index idx generated using this procedure. According to eq. (4), the corresponding MSYT will have entries $1, 2, \ldots, c_1$ at the first column positions given above and blanks at the remaining $N-c_1$ locations. If these blanks are now filled in an increasing sequence with $c_1+1, c_1+2, \ldots, N$ from left to right we obtain the MSYT satisfying the Young rules. As an illustration, consider again the representation $[2^3]$ of $S(6)$. The first column entries generated in an indexed form are,

$$(1,3,5),(1,3,4),(1,2,5),(1,2,4),(1,2,3).$$

The partially filled MSYT that result on using eq. (4) are,

$$[1,-,2,-,3,-],[1,-,2,3,-,-],[1,2,-,-,5,-],$$
$$[1,2,-,3,-,-],[1,2,3,-,-,-].$$

Filling the blanks with entries 4, 5, 6 as outlined above we obtain the MSYT listed earlier for this representation. This technique for generating and indexing the MSYT is reasonably efficient so that it can be done on run time. Thus, we can avoid a $N \times f_{N,\lambda}$ dimensional storage.[4]

This procedure has been programmed and applied, first to generate the representation matrix of dense cycles,

$$C_{(i,j)} = (i, i+1, \ldots, j-1, j). \tag{10}$$

We note that $C_{(i,j)}$ can be represented as a product of ETs, $\{(k,k+1) \mid k = i,\ldots,j-1\}$. Thus, we can obtain the matrix elements, $\langle a|C_{(i,j)}|b\rangle$ for two MSYT by reading the $N \times N$ tables for $\mathrm{diag}(k,k+1)$ and $od(k,k+1)$. In this context, the ordering of entries in MSYT in increasing sequence is important. For the cycle of eq. (10) the matrix element will be zero unless the entry values of $\langle a|$ match with those of $|b\rangle$ for $k = 1, i-1$, and beyond j. Secondly, the entry at $k+1$ of $|b\rangle$ resulting from $(k,k+1)$ should match with the one in $\langle a|$. The matrix element will be zero otherwise, because this entry value will not be affected by other ETs defining $C_{(i,j)}$. Thus, branching under the action of ETs is avoided at the expense of carrying out $f^2_{N,\lambda} \times (j-i)$ operations. The individual operations in the present scheme were, however, found to take 0.06 μs of CPU time on a DEC ALPHA-2100 machine so that we could use this method with reasonable efficiency. The results are summarized in Table I for the singlet representations of $N = 8-18$ electrons. The first column of the table lists the N values considered. The second column lists the dimensionalities of the corresponding representations. The third column gives the CPU time in seconds for generating the required matrix representation.

TABLE I.
CPU Times in Seconds for Representation Matrices of Maximum Length Cycles $C_{(1,N)}$ and Transpositions $T_{(1,N)}$ of $S(N)$ for the Spin Singlet States of N Electrons.

$S(N)$	Dimensionality	CPU Time (Seconds) per Matrix	
N	$f_{(N,\lambda)}$	$C_{(1,N)}$	$T_{(1,N)}$
8	14	0.00018	0.00051
10	42	0.00162	0.0044
12	132	0.0167	0.044
14	429	0.180	0.565
16	1430	1.420	10.420
18	4862	16.50	154.800

Consider now, the transposition, $\{T_{(i,j)},\ j > i\}$, which can be expressed as a product of dense cycles as,

$$T_{(i,j)} = C_{(j,i+1)} \times C_{(i,j)}. \tag{11}$$

Because the ETs $\{(k,k+1) \mid k = 1,\ldots,j-2\}$ are common to both cycles on the right side of eq. (11), the technique for cycles cannot be used directly and matrix multiplication or a recursive scheme becomes unavoidable. We have tried both the approaches and found that matrix multiplication was faster. Hence, the latter approach has been preferred in this note. We have, however, tried to minimize this effort by using the procedure developed for cycles for the forward cycle, $C_{(i,j)}$ of eq. (11). Storing the matrix for the forward part in the core, we express the backward cycle as a product of ETs $(k,k+1)$. The sparse matrix for these is then obtained starting with $k = i+1$ as follows:

If m is an MSYT index of the column of the stored matrix, let $m(k)$, $m(k+1)$ be the relevant entries in it. The diagonal elements of the sparse matrix follow as,

$$\mathrm{matel}(m,m) = \mathrm{diag}\big(m(k)\big). \tag{12}$$

If matel$(m,m) = +1$, or -1, no off-diagonal elements are possible, and we proceed, to consider the next lower ET of the cycle. If not, two possibilities arise,

(i) $m(k) \geq c_1; \quad m(k+1) \leq c_1;$

(ii) $m(k) \leq c_1; \quad m(k+1) \geq c_1.$

Defining,

$$k1 = m(k+1), \tag{13}$$

$$k2 = m(k) - c1 + 2, \tag{14}$$

the off-diagonal MSYT has the index,

$$m1 = m - w(k1, k2) + w(k1, k2 - 1), \tag{15}$$

where ws are weights from the weight diagram[7] for the given representation.

Similarly, for case (ii), defining,

$$k1 = m(k), \tag{16}$$

$$k2 = m(k+1) - c_1 + 1, \tag{17}$$

we obtain,

$$m2 = m - w(k1, k2) + w(k1, k2 + 1). \tag{18}$$

The appropriate off-diagonal matrix element value can be read from the $N \times N$ table. The product of the sparse matrix of the ET and that of $C_{(i,j)}$ are determined and stored in the original area of the former. This procedure is continued until all ETs belonging to $C_{(j,i+1)}$ have been exhausted. The CPU timings (in seconds) for the transpositions are listed in the fourth column of Table I. Repeated calling of the above routine and sparse matrix multiplication leads to the timings being on an average 3 to 10 times slower than for cycles.

As the next part of the present work we consider the DRT approach valid for general representations [λ] of $S(N)$. A geneological realization of this representation is possible if we start with one box corresponding to[1] of $S(1)$ and add boxes one at a time so that at each stage we have only allowed representations of $\{S(i);\ i = 1, \ldots, N\}$. Algebraically, this procedure consists of starting with an array of k zeroes defining the starting representation vector, r, of the $S(0)$ group. If we now increment the entries by unity ensuring that the new arrays define nonincreasing sequences from left to right we obtain the representations of $S(1), \ldots, S(N)$ successively. We also need to ensure that at no stage do the entries exceed in value the corresponding row lengths of the given [λ] of $S(N)$. The set of allowed r-vectors get grouped according to increasing values of N. These k-component arrays are stored in the first k columns of the DRT. We note that the r-arrays of each group $S(i)$ are linked geneologically to corresponding arrays of $S(i+1)$. The linkage occurs whenever the two arrays coincide in all components except in one, say m, in which the difference is unity. These linkages can, in turn, be stored as k-component ln k-arrays and stored in the next k columns of the DRT. The entry values in the corresponding locations of the MSYT now follow on using,

$$l = c_{(m-1)} + r(m), \tag{19}$$

where $c_{(m-1)}$ is the cumulative length up to m as defined by eq. (2) and $r(m)$ is the mth component of the r-vector of $S(i+1)$. The resulting entry values, e-array may be stored in the subsequent k columns of the DRT. Reading the DRT using the

TABLE II.
Distinct Row Table for the Representation [3,2,1] of $S(6)$.

S. No.	r-Vector	ln k-Vector	e-Vector	Cumulative Index	Path Weights
1.	1 0 0	2 3 0	4 2 0	16	0 8 0
2.	1 1 0	4 5 0	6 2 0	8	0 2 0
3.	2 0 0	5 6 0	4 3 0	8	0 6 0
4.	1 1 1	7 0 0	2 0 0	2	0 0 0
5.	2 1 0	7 8 9	6 5 3	6	0 2 4
6.	3 0 0	9 0 0	4 0 0	2	0 0 0
7.	2 1 1	10 11 0	5 3 0	2	0 1 0
8.	2 2 0	10 12 0	6 3 0	2	0 1 0
9.	3 1 0	11 12 0	6 5 0	2	0 1 0
10.	2 2 1	13 0 0	3 0 0	1	0 0 0
11.	3 1 1	13 0 0	5 0 0	1	0 0 0
12.	3 2 0	13 0 0	6 0 0	1	0 0 0
13.	3 2 1	—	—	1	—

link vector components, the MSYT arrays can be readily generated. The determination of the index of a given array follows as in the study by Shavitt[11] using the cumulative path index at each row of the DRT, which may be stored in a further column of the table. The actual path index follows on defining partial indices at each component of the row of DRT. As an illustration we summarize in Table II the construction of the DRT for the representation [3, 2, 1] of the group $S(6)$. The first column of Table II gives the serial number of the row. Columns 2–4 yield the representation components. Columns 5–7 are the components of the ln k-vector. Columns 8–10 yield the components of the e-vector. Column 11 gives the cumulative index at the given row. The last three columns are the partial path indices useful for indexing an MSYT array.

Discussion

Apart from the run-time results presented in Table I and the reduction in memory requirements that result on using MSYT, there are some features that need to be highlighted. First, the present realization emphasizes the location of an entry in the YD and not its value. This is in contrast to the SYT labeling where entries themselves are significant and their locations are specified implicitly by counting them in a row- or column-wise manner. This, as pointed out in the previous section, leads to a given pair of entries having different axial distances, depending on their locations in different tableaux. Second, each entry value of an MSYT can be handled independently of the others. This is what enabled the DRT approach for representations. It would be extremely difficult to implement this scheme using SYT labeling.

We are currently interested in spin-free configuration interaction studies in molecules using spin-adapted basis. This necessitates at most two-column YD representations of $S(N)$. It is for this reason that we did not use the DRT approach to generate the representation matrices.[13] The other reason for not running this program is that the memory requirement for DRT approach is more than that for the graphical one, and so we do not recommend its use unless one is specifically interested in general representations. The run times for generating the MSYT and indexing them are comparable in both approaches. One possible area of application for DRT we are considering is to study spin Hamiltonians for high-spin systems where spin adaptation reduces the dimensionality considerably.

From Table I we note that representations of dense cycles are very easy to generate, and the present approach is quite efficient for this. It may be noted that studies involving effective one-body Hamiltonians lead to only dense cycles as matching permutations. Thus, in these studies the present approach for two-column YD can be readily used. It must, however, be noted that general permutations cannot be readily handled unless we use sparse matrix multiplication techniques for some parts at least. This was clearly brought out in the study of general transpositions.

Acknowledgment

The authors are extremely grateful to IIT-Bombay for providing the necessary infrastructure.

References

1. Hamermesh, M. Group Theory and its Application to Physics; Addison–Wesley: New York, 1963.
2. Kaplan, I. G. Symmetry of Many-Electron Systems; Academic: New York, 1975.
3. Pauncz, R. The Symmetric Group in Quantum Chemistry; CRC Press: London, 1995.
4. Rettrup, S. Int J Quant Chem 1986, 29, 119.
5. Duch, W. Int J Quant Chem 1985, 27, 59.
6. Wormer, P. E. S.; Paldus, J. Int J Quant Chem 1979, 16, 1307; ibid 1979, 16, 1321; ibid 1980, 18, 841;
7. Sarma, C. R.; Ahsan, M. A. H.; Rettrup, S. Int J Quant Chem 1996, 58, 637.
8. Sarma, C. R.; Rettrup, S. Theoret Chim Acta 1977, 46, 63.
9. Sahsrabuddhe, G. G.; Dinesha, K. V.; Sarma, C. R. Theoret Chim Acta 1980, 54, 333.
10. Paldus, J. In Lecture Notes in Chemistry: Unitary Group; Hinze, J., Ed.; Springer Verlag: New York, 1981, p. 1, vol. 22.
11. Shavitt, I. In Lecture Notes in Chemistry: Unitary Group; Hinze, J., Ed.; Springer Verlag: New York, 1981, p. 51, vol. 22.
12. Duch, W.; Karwowski, J. In Lecture Notes in Chemistry: Unitary Group; Hinze, J., Ed.; Springer Verlag: New York, 1981, p. 260, vol. 22.
13. The program segment leading to the DRT and generating indexed MSYT is available from authors, and will be provided on request.